# 基于泊松填充的纹理自适应插值方法 *

张　军[a,b]，陈凯雯[a]

(江南大学 a. 数字媒体学院; b. 江苏省媒体设计与软件技术重点实验室, 江苏 无锡 214122)

**摘　要：** 针对新图形技术条件下的纹理合成问题，提出一种纹理插值方法，可将源纹理图像自适应拉伸为不同尺寸的纹理图像并保持其清晰度不改变。首先，采用高维图像插值算法将源纹理裂变为目标纹理分辨率，作为中间过渡纹理;其次，利用自然图像的自相似性，依据中间纹理像素特征随机从源纹理中选取像素块；最后，使用泊松图像编辑算法将源纹理像素块平滑嵌入到中间纹理的间隙区域，得到最终的合成纹理。通过与现有算法的大量对比实验表明，该算法对静态和非静态纹理合成问题都能适用，且合成结果与源纹理具有较高的视觉一致性。另外，该算法逻辑简单、计算快速，无法复杂优化计算或者学习训练步骤，适合在低硬件配置的移动平台应用。

**关键词：** 纹理合成；高维插值；泊松方程；非静态纹理

**中图分类号：** TP391.41　　doi: 10.19734/j.issn.1001-3695.2018.10.0778

# Adaptive texture interpolation by Poisson filling

Zhang Jun[a, b], Chen Kaiwen[a]

(*a. School of Digital Media*, *b. Jiangsu Key Laboratory of Media Design & Software Technology*, *Jiangnan University*, *Wuxi* Jiangsu 214122, *China*)

**Abstract:** Aiming at the texture synthesis in new computer graphics technical conditions, this paper proposes a texture interpolation method to synthesize different resolution textures adaptively from a single real world texture. The synthesized texture's DPI is same as the original texture example. Firstly, the proposed method use a high-dimensional interpolation algorithm to generate an intermediate guidance texture with target resolution by split the source texture. Secondly, particular random patches are selected to fill the gaps in the intermediate texture according to the self-similarity in the source texture. Finally, these patches are seamlessly embedded in the previous intermediate texture by the Poisson image-editing algorithm. Experimental results show that the proposed method can handle both stationary and non-stationary texture synthesis, and the synthetic results are more consistent in the visual properties of the source texture. In addition, the implement logic of the proposed method is enough simple to be programing and executing on a common mobile platforms.

**Key words:** texture Synthesis; high-dimensional interpolation; Poisson filling; non-stationary texture

## 0　引言

近些年，电影、游戏等行业对计算机图形的逼真度需求不断攀升，对三维虚拟场景的视觉真实性提出越来越高的标准。然而，图形硬件的瓶颈使业界无法通过提高几何网格数量和光照计算复杂度进一步增加虚拟场景真实性，只有通过大量使用高质量纹理贴图的方式间接增强场景真实感。

使用纹理贴图的一个突出问题是在面对不同显示区域大小时，纹理贴图经常需要不同尺寸的图像。为避免存贮大量高分辨率图像，纹理合成技术可以使用少量样例或参数，以程序化方式动态生成所需纹理，得到业界普遍认可。因此，现有文献里存在大量纹理合成技术研究，在计算机图形学应用领域取得了令人瞩目的成就[1, 2]。当前主流纹理合成算法主要采用纹理像素块拼接（patch-based matching）的思想，根据各类选取准则，从样例图像中选取子区域像素块复制到目标纹理中[3~10]。为保证合成纹理在视觉上没有重复感和断裂感，许多优化“块”选择或排列的算法被相继提出[11-15]。这些算法将纹理块库视为自变量定义域，将拼接效果定义成一个误差能量函数，从而将纹理合成问题归结为求解能量最小值的传统优化问题。这样的思路在面对小尺寸样本和具有高度重复性的静态纹理时，取得了较好的合成效果，可生成人工痕迹很小的大尺寸纹理。

对于高分辨率的非静态纹理样本，其几何特征存在多个尺度上的空间异构分布，现有主流纹理合成技术无法将所有特征无缝的拼接到目标纹理中。针对这一困难，近两年有学者相继提出追加纹理特征参考图（guidance map）的形式[16]和基于深度学习形式[17-19]的解决方案。然而，现有特征参考图方法[16]中纹理特征分析过程涉及复杂非线性降维计算，应对高分辨率样例纹理时其计算速度令人难以接受。基于对抗性深度学习[18]的方法虽然能快速生成高分辨率纹理，但其学习、训练过程时间漫长且学习后的网络数据量庞大，实现程序需要非常大的内存用来存储训练数据集和训练结果，基本无法应用到移动平台。另外，深度学习型算法一旦完成训练后，缺乏可调参数用来满足不同纹理合成需求，只有更新学习样例库重新训练，缺乏灵活性。

实际上，随着计算机硬件和图像采集技术的发展，工程中对合成无限大小人工纹理的需求逐渐减少，较多应用场景是对已有高分辨率纹理进行缩放或拉伸操作，以适应不同屏

**收稿日期：** 2018-10-09；**修回日期：** 2018-11-07　**基金项目：** 国家科技支撑计划资助项目（2015BAH54F01）；江苏省自然科学基金资助项目（BK20141113，BK20130158）

**作者简介：** 张军，副教授，博士，主要研究方向为图像处理、计算机仿真技术、可视化技术、航空电子（zhangjunman@jiangnan.edu.cn）；陈凯雯，硕士研究生，主要研究方向为计算机图形学、计算机仿真技术.

幕尺寸（手机、平板、VR/AR 设备、可穿戴设备、PC 等）的需求。为保证计算机游戏、动画电影等虚拟场景在不同设备上的视觉一致性，要求合成后的纹理必须与源纹理保持一致的视觉特征和清晰度（dots per inch，DPI）。

本文作者将这种新的纹理合成需求归结为纹理插值问题，提出一种高维图像块插值算法生成中间纹理，再采用泊松图像编辑方法逐步填充像素块之间的间隙。该算法可适用于非静态纹理自适应缩放的应用，在保持源纹理特征分布和 DPI 不变的情况下，合成指定分辨率的纹理图像（如图 1）。由于未采用复杂优化或学习型方法，该算法具有线性时间复杂度 $O(N)$，并具有简单、易于编程、对硬件需求低的优势。



图 1　本文算法对源纹理图像（中间排）的插值结果（上、下排）

Fig. 1　Two interpolation results (upper and lower rows) from the original textures (middle row) by the ATIPF

需要特别指出的是，传统图像重映射[20-22](Image Retargeting)或图像超分辨率[23, 24]方法并不适用于纹理插值问题。现有图像重映射方法本质上是图像自适应裁剪算法，较少具备图像放大能力。而图像超分辨率方法会严重影响源纹理的 DPI，造成视觉质量的损失。

## 1　算法框架

本节从总体算法流程上介绍基于泊松填充的纹理插值方法，其理论细节将在第 3 节中展开。基于样例的纹理合成技术发展至今，大体可分为基于块拼接的方法和基于像素合成的方法。由于块拼接可以保持合成结果的 DPI 与样例图像一致，所以更适合大范围纹理合成的需求。

基于样例图像的块拼接合成技术的主要操作步骤，是从样例图像中选取合适的块（比如 8×8 区域）写入到合成纹理图像的特定位置（如图 2）。为了让合成纹理视觉上不出现拼接痕迹，需要精心设计块的选择方式和像素写入算法。

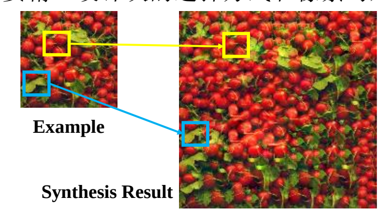



图 2　基于块拼接的纹理合成算法原理

Fig. 2　Texture synthesis algorithm based on patches splicing

与传统块拼接合成技术不同，本文算法不但保持合成结果不出现拼接痕迹，还需要保持原样例图像的各个尺度的特征分布。Zhou 等人[16]的纹理分析方法可以对样例特征分布进行分析，得到一幅“特征参考图”用来引导纹理块的选择，从而保持纹理特征分布不改变。此方法涉及非线性维数约减计算，在面对高分辨样例图像时，计算极其缓慢并消耗大量内存空间。本文采用图像插值的思想，并将其推广到高维纹理像素块的插值情形，从而可以快速得到与原样例纹理特征分布一致的中间骨过渡纹理。该方法先将样例图像分裂成块状，并均匀散布到目标纹理图像中，再使用高维插值填充块与块之间的间隙。

在过渡纹理中，块与块间隙位置的像素是通过高维插值得到的，具有较为明显的拼接痕迹和不自然的镜像特性。为此，本文算法利用自然纹理的自相似特性，随机从样例图像中寻找与过渡纹理中块与块的间隙位置图像特征相似的像素块，将其嵌入到该间隙位置，从而消除中间纹理中高维插值带来的人工拼接痕迹和不自然性的镜像特性。

最后，根据参考图像的纹理特征信息，逐个从源为图像中寻找最接近参考图像的像素块，并用泊松图像编辑方法填充到各个间隙位置，形成最终的合成纹理。

本文算法（Adaptive Texture Interpolation by Poisson Filling，**ATIPF**）流程图如图 3 所示，该算法仅涉及图像分割、翻转、比较等计算，编程实现相对简单，不涉及复杂网络关系图或学习型计算。

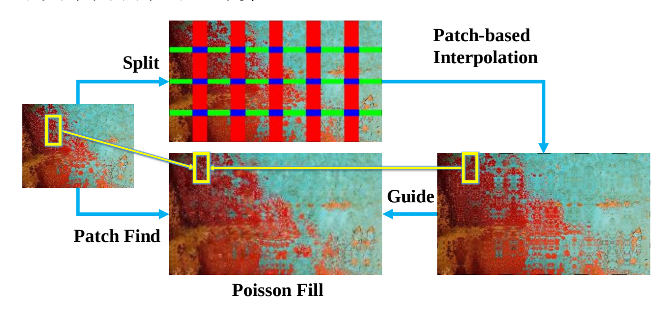



图 3　ATIPF 算法流程图

Fig. 3　Outline of the ATIPF

ATIPF 算法描述如下：

***Input***：*texture* $\mathbf{I}$ , *patch size* $p$ , *output size* $m', n'$

***Output***: *synthesized texture* $\mathbf{I}'$

1. Crack texture $\mathbf{I}$ to $p \times p$ blocks $\mathrm{B}_{ij}$ and locate the gaps $\mathrm{LR}_{ij}$ $\mathrm{UD}_{ij}$ $\mathrm{C}_{ij}$
2. Do interpolation:
$$\mathrm{LR}_{ij} = \mathrm{interp}\ \ B_{ij-1}, B_{ij+1}$$
$$\mathrm{UD}_{ij} = \mathrm{interp}\ \ B_{i-1j}, B_{i+1j}$$
$$\mathrm{C}_{ij} = \mathrm{interp}\ \ UD_{i-1j}, UD_{i+1j}, LR_{ij-1}, LR_{ij+1}$$
3. Find optimal gaps
$$\mathrm{LR}_{ij} \leftarrow \mathrm{opt}(\mathrm{I}, \mathrm{LR}_{ij})$$
$$\mathrm{UD}_{ij} \leftarrow \mathrm{opt}(\mathrm{I}, \mathrm{UD}_{ij})$$
$$\mathrm{C}_{ij} \leftarrow \mathrm{opt}(\mathrm{I}, \mathrm{C}_{ij})$$
4. Write blocks to output texture $\mathbf{I}' \leftarrow \mathrm{B}_{ij}$
5. Fill gaps to output texture by Poisson edition
$$\mathbf{I}' \xleftarrow{\text{Poisson Edition}} \mathrm{LR}_{ij}\ \ \mathrm{UD}_{ij}\ \ \mathrm{C}_{ij}$$

在 **ATIPF** 算法中，第 1 至 2 步为“特征参考图”生成过

程，其计算为图像块间的线性插值计算，计算量与目标纹理图像的像素个数成正比。算法的第 3 至 4 步为寻找与参考图像最接近的像素块计算，是从源纹理图像中随机选取固定数量像素块与参考图像的像素块进行欧氏距离计算，其计算量与参考图像中的像素块数量成正比，从而与目标纹理图像的像素量成正比。算法的第 4 至第 5 步为像素块写入目标纹理图像计算，在采用固定像素尺寸和固定迭代步数的 **Jacobi** 方法求解泊松方程时，其计算量同样与像素块的数量成正比，从而与目标纹理图像的像素量成正比。最终，**ATIPF** 的计算量与目标纹理图像的像素量成正比关系，其计算复杂度为 $O(N)$。

另外，**ATIPF** 计算流程中具有高度的像素块独立计算特性，适合采用并行计算策略，具备在多核 CPU 或 GPU 上采用并行计算方案的潜力。

## 2　高维图像块插值

本文关注的纹理合成技术需要对原始纹理样例的各种尺度特征给予保留，即要保持纹理宏观特征分布不发生改变，也要保持为微观特征不被丢弃。Zhou 等人[16]的纹理分析方法可以给样例纹理的各个尺度特征赋值一个标量参数，从而指导纹理合成算法在挑选纹理块的时候选择特征一致的区域。但 Zhou 等人[16]的纹理分析方法需要复杂的非线性归约计算，对高分辨率图像的处理速度非常缓慢，并不能满足本文快速计算的需求。

本节提出一种图像插值技术，可以快速生成中间纹理图像，并以此作为纹理合成时像素块选择时的依据。其思想是传统像素级插值算法的推广，将低维像素级插值问题推广至高维像素块的插值问题。

图像插值问题已经得到了广泛的研究，但基本都是基于像素级的插值思想，其经典方案就是双线性插值（如图 4）。其思想是在原始图像的像素点之间插入新点 $Q_{ij}$，并与周边像素点特性尽量保持一致：

$$Q_{ij}=u\left[vP_{ij}+(1-v)P_{ij+1}\right]+(1-u)\left[vP_{i+1j}+(1-v)P_{i+1j+1}\right] \tag{1}$$

图 4　经典双线性图像插值算法原理示意图

Fig. 4　Principle of the bilinear interpolation algorithm

经典双线性图像插值算法因为采用了像素级的计算策略，从而破坏了原始图像的 DPI，使插值结果总有一定的模糊现象。本节提出基于像素块的插值思想，在高维空间上对纹理图像进行插值，并保持较快的计算速度。

如果将像素点 $\{P_{ij}\}$ 推广至像素块 $\{B_{ij}\}$，先将源纹理分裂成规则图像块，并均匀散步在待插值纹理图像上（如图 5），则图像插值问题依然可以用式（1）的形式。

$$C_{ij}=u\left[vB_{ij}+(1-v)B_{ij+1}\right]+(1-u)\left[vB_{i+1j}+(1-v)B_{i+1j+1}\right] \tag{2}$$

实验结果显示，式（2）形式的图像插值算法，会因为图像块所带有的边缘信息而产生人眼非常容易注意到的块状现象，并不能直接应用。

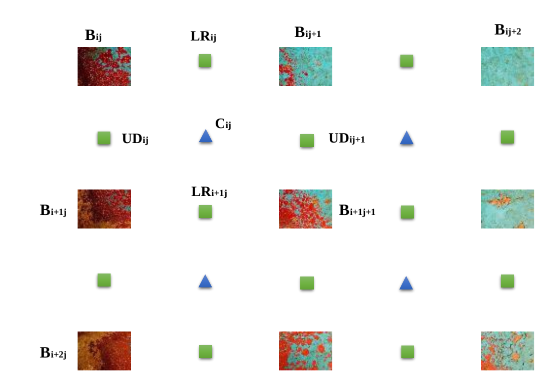



图 5　基于图像块的高维图像插值原理示意图

Fig. 5　Principle of the high-dimensional interpolation by image patches

为此，本节提出类似径向基函数的形式[25-27]，采用镜像映射和过渡函数的形式插值出平滑过渡的无块状效应的纹理图像。为方便叙述，本节将图 5 中的间隙按照与源纹理图像块的位置关系分为 $\{LR_{ij}\}\{UD_{ij}\}\{C_{ij}\}$ 三种类型。

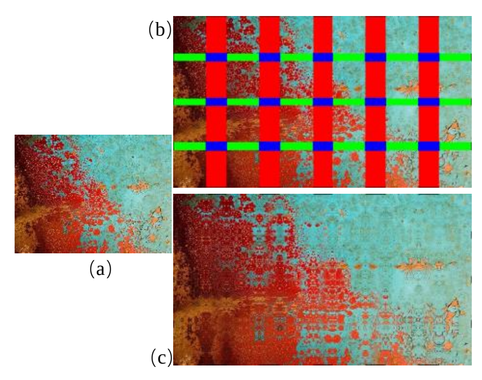
记符号 lrflip 为水平镜像映射（左右翻转），udflip 为垂直镜像映射（垂直翻转），则本节提出的高维纹理插值计算公式为：

$$LR_{ij}=\sin^2\left(\tfrac{x_{ij}}{N}\right)lrflip\left(B_{ij-1}\right)+\cos^2\left(\tfrac{x_{ij}}{N}\right)lrflip\left(B_{ij+1}\right)$$

$$UD_{ij}=\sin^2\left(\tfrac{y_{ij}}{M}\right)udflip\left(B_{i-1j}\right)+\cos^2\left(\tfrac{y_{ij}}{M}\right)udflip\left(B_{i+1j}\right)$$

$$C_{ij}=\tfrac{1}{4}\left(UD_{i-1j}+UD_{i+1j}+LR_{ij-1}+LR_{ij+1}\right) \tag{3}$$

由于过渡基函数在像素块边缘是平滑衔接的，式（3）的插值结果无明显块状痕迹，仅有镜像引起的非自然效果。考虑到本节图像插值算法的目标是为寻找纹理块提供依据和准则，其效果完全符合纹理插值的需求。




(a)源纹理图像　(b)分块填充结果　(c) 高维插值填充间隙的结果

(a) Source texture (b) Split result (c) Filled by high-dimensional interpolation

图 6　高维图像插值计算结果

Fig. 6　The result of high-dimensional interpolation

## 3　泊松填充

本文第三节提出的图像高维插值方法虽然可以达到无块状拼接痕迹的纹理图像缩放效果，但其插值结果存在较多镜像的非自然区域。对于纹理复杂的且周期性很强的纹理图像来说，这种镜像效果不明显。但对于一般非周期性纹理，特别是非静态纹理，这种效果很容易被人眼察觉。但该计算结

果可以作为中间纹理，指导纹理块寻找算法从源纹理图像中得到与中间纹理最为近似的像素块，用于替换间隙区域的像素块。

在替换间隙时，不能破坏间隙与源纹理块之间的平滑过渡，这就需要寻找合适的图像块融入方法。传统纹理合成算法已经提出了许多类似的算法，其效果在静态纹理合成问题上较为成功。本文作者为避免复杂优化计算或者边界优化分割等耗时的计算过程，选择泊松图像编辑算法[28]作为像素块融入中间纹理间隙的手段。

泊松图像编辑算法的原理较为简单，其核心思想是在区域 Ω 上求解泊松方程：

$$\Delta f(x,y) = g(x,y) \tag{4}$$

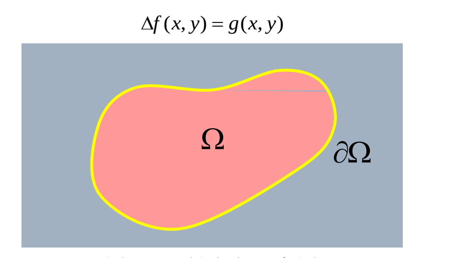



图 7　泊松方程示意图

Fig. 7　Sketch of the Poisson equation

其中 $I = f(x,y)$ 是待求解的图像块，$\Delta \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ 是拉普拉斯算子。设从源纹理中寻找到的最接近间隙块 $\mathrm{LR}_{ij}$（仅以此类型间隙为例，其他类型间隙的计算过程与其一致）的纹理块为 $\mathrm{LR}_{ij}^{opt}$，可将其拉普拉斯变换结果及其边界条件代入方程（4），从而完成泊松方程的建立：

$$\begin{cases} \Delta f(x,y) = \mathrm{LR}_{ij}^{opt}(x,y) \\ \partial f(x,y) = \partial \mathrm{LR}_{ij}^{opt}(x,y) \quad \text{with } (x,y) \in \partial\Omega \end{cases} \tag{5}$$

最优纹理块可采简单的采用像素平方距离定义：

$$\mathrm{LR}_{ij}^{opt} = \arg\min_{\mathrm{B}_k} \|\mathrm{LR}_{ij} - \mathrm{B}_k\| \tag{6}$$

其中 $\mathrm{B}_k \subset I$ 是源样例纹理中任意一个子区域的图像块，在实际计算时，本文采用蒙特卡罗法随机抽取 $K$ 个图像块样本，作为方程（6）的搜索空间。



图 8　泊松填充步骤的计算结果

Fig. 8　Final result by the Poisson image-editing

从计算结果图 8 可以看出，经过泊松填充的纹理图像已经消除了中间纹理的镜像效果，但充分保留了中间纹理的特征分布情况，从而维持了源样例纹理的特征分布情况。进一步，由于泊松填充所采用的所有纹理块均直接来自源纹理图像，未经其他图像处理步骤，故其 DPI 与源纹理保持一致，达到拉伸源纹理而保持其清晰度不变的目标。

## 4　计算结果及讨论

本章给出与同类经典算法对比的实验结果，用以证明本文 ATIPF 算法可以对静态和非静态纹理图像进行快速插值计算，具有较高自适应性和鲁棒性，更适合纹理图像的自适应缩放应用场景。

所有实验结果均在一个搭载有 3.4 GHz Intel CPU 的 PC 平台上得到，实验编程环境为 Matlab 2015b，前人算法的实验结果皆为其论文作者公布的计算结果。为便于描述，本节涉及到的前人算法简称如表 1 所示。

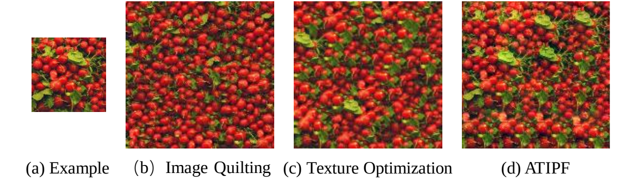
表 1　算法简称与对应文献编号表

Table 1　The corresponding of the abbreviation in experimental results

| 简称 | Image Quilting | *Texture Optimization* | *Image Melding* | *Graphcut Textures* | *Self Tuning* | *DeepCorNTSAE* | |
|---|---|---|---|---|---|---|---|
| 文献 | 文献[4] | 文献[12] | 文献[14] | 文献[13] | 文献[11] | 文献[17] | 文献[18] |

### 4.1　静态纹理插值效果对比实验

对于静态纹理，源纹理图像中包含较多同一尺度的重复纹理特征，其合成结果大多为随机抽取源纹理块状像素信息，再拼接到目标纹理中。为保证合成后的纹理没有明显的周期性重复或者拼接痕迹，现有算法采用最优边界切割（image quilting）[4]或者最优块区域选择（texture optimization）[12]的策略。



(a) Example　(b) Image Quilting　(c) Texture Optimization　(d) ATIPF

图 9　静态纹理对比实验

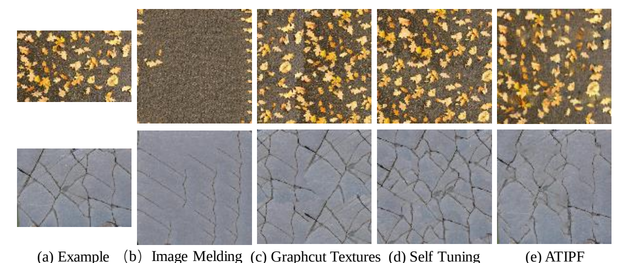
Fig. 9　The synthetic results on static textures

从实验对比结果（图 9）中可以看出，静态纹理合成相对是一个较为简单的问题，经典算法大多能有效合成无拼接痕迹的纹理。由于采用了高维图像插值结果作为纹理合成依据，本文算法的优势在于保持源纹理的特征分布规律。例如，图 9c 中仅含有少量大片绿叶区域，并且其分布位置与源图像也不一致。

### 4.2　非静态纹理插值效果对比实验

非静态纹理一般是自然照片类纹理，存在多种尺度结构特征，具有非周期、非规则性的纹理空间分布特性，是经典纹理合成算法无法适应的一类源纹理。另一方面，自然照片一般具有较高分辨率，会大幅加大优化类算法的搜索空间，使其计算效率迅速下降。

Kaspar 等人[11]提出的 Self Tuning 算法可以适应弱非静态纹理合成的场景，但其多尺度优化计算过程耗时过长，且对强非静态纹理合成并不适用。Zhou 等人[16]提出“Guidance Map”的方式指导 Self Tuning 算法的非静态纹理合成过程，取得较好效果。但其 Guidance Map 的计算过程涉及非线性降维计算，在处理高分辨源纹理时非常耗时。本节将给出本文 ATIPF 算法与 Self Tuning 等算法的对比结果。



(a) Example　(b) Image Melding　(c) Graphcut Textures　(d) Self Tuning　(e) ATIPF

图 10　非静态纹理对比实验 1

Fig. 10　The synthetic results on non-static textures I

图 10 给的是本文算法对简单的非静态纹理源图像的合成效果实验对比结果，其中图 10b 是 Image Melding[14]方法合成效果，图 10c 是 Graphcut Textures[13]方法合成结果，图 10d 是 Self Tuning[11]方法合成结果，图 10e 是本文 ATIPF 算法合成结果。

未经全局优化的纹理合成算法容易重复选择纹理接近的像素块嵌入目标纹理中，最终合成过于平坦（图 10（b））或者有明显的背景分割痕迹（图 10（c））的结果。Self Tuning[11]方法通过复杂的全局优化方式强制合成纹理均匀的选取源纹理中的所有图像块，从而达到了较好的合成效果，但无法做到纹理特征分布与源纹理保持一致（图 10d）。本文提出的 ATIPF 算法无须全局优化计算，通过纹理插值的方式引导像素块选择过程，从而生成与源纹理特征分布一致的合成纹理（图 10（e））。

需要注意的是，本文 ATIPF 算法采用了泊松融合算法向合成纹理嵌入图像块，有可能在合成纹理的少量分块边界域存引入一定颜色“晕染”现象（图 10（e）上排）。

## 4.3 复杂纹理插值效果实验

对于复杂的非静态纹理，传统基于块拼接的算法缺乏对源纹理各个尺度特征的描述，即使采用全局优化搜索方法[11]也无法达到令人满意的结果。为此，Zhou 等人[16]提出用“Guidance Map”指导纹理块搜索的策略，虽然一定程度上改善了合成结果，但其涉及复杂非线性降维计算的缺陷使之无法适用高分辨率源纹理的应用场景。最近，Zhou 等人[18]继续提出深度学习型纹理合成算法，其针对复杂非静态纹理的合成结果得到大幅改善。

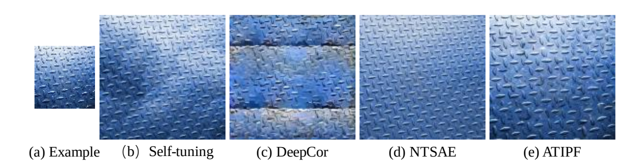



图 11　非静态纹理对比实验 2

Fig. 11　The synthetic results on non-static textures II

图 11（a）的源纹理中包含两个尺度的问题特征，铁板上钉状突起按照两个方向，铁板本身因光照而产生不均匀的亮度分布。Self Tuning[11]方法通过全局优化搜索，适应了源纹理的钉状突起特征分布，但改变了其背景亮度特征分布（图 11（b））。DeepCor[17]方法无法适应复杂非静态纹理问题，其合成结果（图 11（c））基本失去了纹理合成作用。NTSAE[18]方法的合成结果成功适应了多尺度纹理特征的非均匀性，其合成结果视觉一致性最高。本文 ATIPF 算法的合成结果算存在少量钉状突起的重叠，但其背景亮度分布与源纹理高度一致，在保留源纹理特性方面优于 NTSAE[18]这种复杂深度学习方法，在实用上更具竞争力。

图 12 给出本文 ATIPF 算法对高分辨照片纹理（1024×600）的插值结果，可以看出，对于复杂的自然纹理，源纹理自身存在较多自相似性，可被本文算法充分利用而产生与源纹理具有高度视觉一致性的插值纹理。

虽然 Zhou 等人[18]提出的 NTSAE 学习算法能在很多复杂纹理（如树叶、树木年轮、孔雀尾等）图像合成上取得了令人惊讶的成功，但其合成结果在图像边界部位总带有模糊的效果，图 13 和图 14 给出该现象比较严重的两个案例，体现出该算法鲁棒性不足的一面。

本文提出的 ATIFP 算法具有较好鲁棒性，在充分利用自然图像自相似性的基础上，可以有效应对众多复杂纹理插值场景。

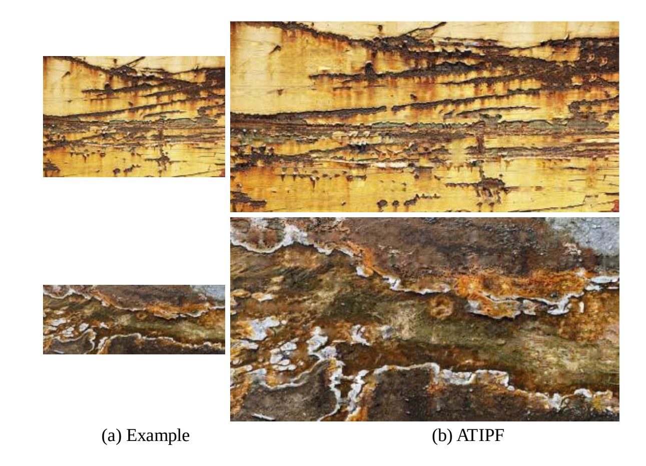



图 12　复杂自然纹理插值实验结果

Fig. 12　The synthetic results on complex nature textures

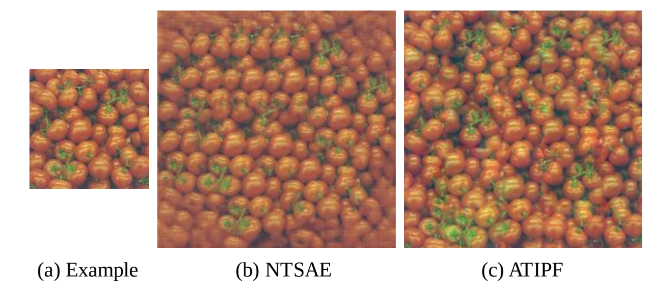



图 13　本文 ATIPF 算法与 NTSAE 算法对比实验结果（一）

Fig. 13　The synthetic result by ATIPF vs NTSAE I

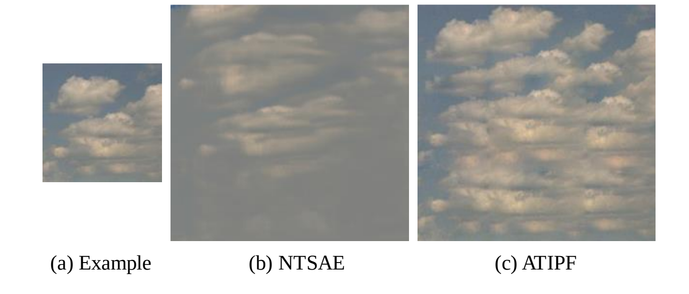



图 14　本文 ATIPF 算法与 NTSAE 算法对比实验结果（二）

Fig. 14　The synthetic result by ATIPF vs NTSAE II

# 5　结束语

本文在图像高维插值和泊松填充的基础上，提出一种能保持图像清晰度不变的纹理插值算法，可自适应的将源纹理图像拉伸到目标尺寸。该算法不涉及复杂全局优化计算或者深度学习训练，对内存要求较低且易于并行化计算，适合包括移动平台在内的普通硬件平台实现。

与传统纹理合成算法相比，本文算法鲁棒性较强，对静态和非静态源纹理图像都有较好的合成效果。另外，本文算法的合成结果在各个尺度纹理特征的分布上都能保持与源纹理一致，非常适合自然纹理图像的缩放、拉伸等应用场景。

最后，对某些存在旋转特征重复的源纹理图像，本文算法的合成效果劣于基于深度学习的 NTSAE 算法，需要寻找更合适的像素块搜索方案。

## 参考文献：

[1] Lagae A, Lefebvre S, Cook R, *et al.* A survey of procedural noise functions [J]. Computer Graphics Forum. 2010, 29 (8): 2579-2600.

[2] Barnes C, Zhang F. A survey of the state-of-the-art in patch-based

synthesis [J]. Computational Visual Media. 2017, 3 (1): 3-20.

[3] Lasram A, Lefebvre S. Parallel patch-based texture synthesis [C]//Proc of the 12th Conference on High-Performance Graphics. Goslar: Eurographics Association, 2012: 115-124.

[4] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer [C]//Proc of the 28th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 2001: 341-346.

[5] Efros A A, Leung T K. Texture synthesis by on-parametric sampling [C]//Proc of the 17th IEEE International Conference on Computer Vision.1999: 1033-1038.

[6] 李扬，吴敏渊，颜佳. 基于改进 PatchMatch 的自相似性图像超分辨率算法 [J]. 计算机应用研究, 2018, 35 (4): 1231-1235. (Li Yang, Wu Minyuan, Yan Jia, Self-similarity based image super-resolution algorithm using optimized PatchMatch [J] .Application Research of Computers, 2018, 35 (4): 1231-1235. )

[7] 沈哲，王莉莉. GPU 适用的并行纹理合成算法 [J]. 计算机辅助设计与图形学学报,2015, 27 (2): 330-336. (Shen Zhe, Wang Lili, Parallel algorithm for texture synthesis on GPU [J] . Journal of Computer-Aided Design & Computer Graphics, 2015, 27 (2): 330-336. )

[8] 韩建伟，王青，周昆，等. 基于 Wang Tiles 的几何纹理合成 [J]. 软件学报,2009, 20 (12): 3254-3264. (Han JianWei, Wang Qing, Zhou Kun, *et al.* Wang tile based geometric texture synthesis [J]. Journal of Software, 2009, 20 (12): 3254-3264. )

[9] 邹昆，韩国强，李闻，等. 基于 Graph Cut 的快速纹理合成算法 [J]. 计算机辅助设计与图形学学报,2008, 20 (5): 652-658. (Zou Kun, Han Guoqiang, Li Wen, *et al.* An efficient method of texture synthesis based on graph cuts [J]. Journal of Computer Aided Design & Computer Graphics,2008, 20 (5): 652-658. )

[10] 徐晓刚，鲍虎军，马利庄. 纹理合成技术研究 [J]. 计算机研究与发展,2002, 39 (11): 1405-1411. (Xu Xiaogang, Bao Hujun, Ma Lizhuang. Study on texture synthesis [J]. Journal of Computer Research and Development,2002, 39 (11): 1405-1411. )

[11] Kaspar A, Neubert B, Lischinski D, *et al.* Self-tuning texture optimization [J]. Computer Graphics Forum, 2015, 34 (2): 349-359.

[12] Kwatra V, Essa I, Bobick A, *et al.* Texture optimization for example-based synthesis [J]. ACM Trans on Graphics,2005, 24 (3): 795-802.

[13] Kwatra V, Dl Sch O A, Essa I, *et al.* Graphcut textures: image and Vvideo synthesis using graph cuts [J]. ACM Trans on Graphics,2003, 22 (3): 277-286.

[14] Darabi S, Shechtman E, Goldman D B, *et al.* Image melding: combining inconsistent images using patch-based synthesis [J]. ACM Trans on Graphics,2012, 31 (4): 82.

[15] Lefebvre S, Hoppe H. Appearance-space texture synthesis [J]. ACM Trans on Graphics,2006, 25 (3): 541-548.

[16] Zhou Y, Shi H, Lischinski D, *et al.* Analysis and controlled synthesis of inhomogeneous textures [J]. Computer Graphics Forum.,2017, 36 (2): 199-212.

[17] Sendik O, Cohen-Or D. Deep correlations for texture synthesis [J]. ACM Trans on Graphics,2017, 36 (5): 161.

[18] Zhou Y, Zhu Z, Bai X, *et al.* Non-stationary texture synthesis by adversarial expansion [J]. ACM Trans on Graphics,2018, 37 (4): 49.

[19] 余思泉，韩志，唐延东，等. 基于对抗生成网络的纹理合成方法 [J]. 红外与激光工程,2018, 47 (2): 34-39. (Yu Siquan, Han Zhi, Tang Yandong,*et al.* Texture synthesis method based on generative adversarial networks [J]. Infrared and Laser Engineering,2018, 47 (2): 34-39. )

[20] Lin S, Yeh I, Lin C, *et al.* Patch-based image warping for content-aware retargeting [J]. IEEE Trans on Multimedia, 2013, 15 (2): 359-368.

[21] 岳朋飞，王化雨，郑元杰，等. 结合边缘模糊的景深显著性在图像缩放中的研究 [J]. 计算机辅助设计与图形学学报,2018, 30 (3): 415-423. (Penfei Yue, Huayu Wang, Yuanjie Zhen, *et al.* Image retargeting using blur based depth saliency descriptor [J]. Journal of Computer Aided Design & Computer Graphics,2018, 30 (3): 415-423. )

[22] 谷香丽，迟静，张彩明. 基于弹簧变形模型的图像缩放方法 [J]. 中国图象图形学报,2018, 23 (5): 756-765. (Gu Xiangli, Chi Jing, Zhang Caiming. Image scaling based on the spring deformation model [J]. Journal of Image and Graphics.,2018, 23 (5): 756-765. )

[23] 刘村，李元祥，周拥军，等. 基于卷积神经网络的视频图像超分辨率重建方法 [J/OL]. 计算机应用研究，2019, 36 (4) . http://www. arocmag. com/article/02-2019-04-057. html. (Cun Liu, Yuanxiang Li, Yongjun Zhou, *et al.* Video image super-resolution reconstruction method based on convolutional neural network [J/OL]. Application Research of Computers，2019, 36 (4) . http://www. arocmag. com/article/02-2019-04-057. html)

[24] 聂栋栋，马利庄. 能量均衡原则下的快速图像尺寸调整算法 [J]. 计算机辅助设计与图形学学报. 2014, 26 (8): 1314-1319. (Nie Dongdong, Ma Lizhuang. Fast image resizing method under energy balance principle [J]. Journal of Computer Aided Design & Computer Graphics. 2014, 26 (8): 1314-1319. )

[25] Zhixiang C, Feilong C. Spherical scattered data quasi-interpolation by gaussian radial basis function [J]. Chinese Annals of Mathematics: Series B. 2015 (3): 401-412.

[26] Carr J C, Beatson R K, Cherrie J B, *et al.* Reconstruction and representation of 3D objects with radial basis functions [C]// Proc of the 28th Annual Conference on Computer Graphics and Interactive Techniques.New York: ACM Press, 2001: 67-76.

[27] Beatson R K, Light W A, Billings S. Fast solution of the radial basis function interpolation equations: domain decomposition methods [J]. SIAM Journal on Scientific Computing. 2000, 22 (5): 1717-1740.

[28] Rez P E, Gangnet M, Blake A. Poisson image editing [J]. ACM Trans on Graphics,2003, 22 (3): 313-318.